Parameter Tuning and Scientific Testing in Evolutionary Algorithms

Smit, S.K.

2012

document version
Publisher's PDF, also known as Version of record

Parameter Tuning and Scientific Testing in Evolutionary Algorithms

Selmar Kagiso Smit

Vrije Universiteit, Amsterdam

Binnen algoritmes spelen parameter een belangrijke, maar vaak onderschatte rol. De instellingen van de parameter bepalen namelijk in grote mate de werking en daarmee de uitkomst van het algoritme. Je zou zelfs kunnen stellen dat ze bijna belangrijker zijn dan het model zelf, immers zonder goede instellingen werkt geen elke algoritme goed. Toch wordt dit tegenwoordig nog geregeld over het hoofd gezien en richt men zich vooral op het ontwikkelen van nieuwe algoritmes, in plaats van betere parameters. In dit proefschrift staat dan ook het *hoe* en *waarom* van het instellen van parameters centraal, specifiek binnen het vakgebied evolutionary algorithms. Evolutionary algorithms, parameter tuning en scientific testing blijken onlosmakelijk met elkaar verbonden.

Evolutionary Algorithms

Een evolutionair algoritme is een speciale manier voor het oplossen van problemen, dat door middel van willekeurige acties een oplossing probeert te vinden. Denk bijvoorbeeld aan een navigatiesysteem dat de snelste route moet vinden van A naar B. Hoewel dat simpel lijkt, aangezien je kunt berekenen hoe lang een bepaalde route duurt, is het behoorlijk inwikkeld om op te lossen. Het probleem is daarbij dat er oneindig veel verschillende mogelijkheden zijn, dus het is onmogelijk om ze allemaal af te gaan. In plaats daarvan kan je beter een manier verzinnen waardoor je altijd binnen een paar seconden een hele snelle (niet noodzakelijk de snelste) route kunt vinden. Een mogelijke manier is bijvoorbeeld om route vanaf het begin op te bouwen, door eerst simpelweg bij A te beginnen. Bij de eerste kruising die je tegenkomt kun je vervolgens op een slimme manier bepalen welke van de vier wegen je kiest. Je zou bijvoorbeeld in 50% van de gevallen kunnen kiezen voor de weg die de goede richting op gaat, in 40% van de gevallen te kiezen voor de weg die richting een snelweg gaat, en in 10% van de gevallen een willekeurige weg te kiezen. Bij de eerstvolgende kruising kan je weer precies hetzelfde doen, en hetzelfde voor alle andere kruisingen die je tegenkomt, net zolang totdat je B hebt bereikt. Een computer kan een dergelijke berekening binnen een fractie van een seconde doen, dus er is tijd genoeg om het geheel een paar honderd keer te doen. De beste van die honderden gevonden routes

zal het navigatiesysteem je dan aanbevelen. Hierdoor heb je dan weliswaar niet altijd de snelste route (want dat zou betekenen dat je precies elke keer goed hebt gekozen), maar in ieder geval heb je wel een binnen een paar seconden een redelijke goede route.

Een evolutionair algoritme werkt zelfs nog iets slimmer. In plaats van honderden keren vanaf het begin te beginnen, begint een evolutionair algoritme een aantal, bijvoorbeeld 50 keer vanaf het begin. Van deze 50 gaat hij eerst bekijken wat de snelste route was. Deze snelste route gebruikt hij als basis, om hopelijk een nog betere route te vinden. De eerste stap is om uit deze route willekeurig een kruispunt te kiezen. Vanaf dit kruispunt gaat de nieuwe route afwijken van de oude, want hij kiest speicifiek een andere weg dan hij eerst had gekozen. Op het moment dat hij dan weer een nieuw kruispunt bereikt, doet hij weer precies hetzelfde als voorheen: met 50% kans de ene kant, met 40% kans de andere kant, en met 10% kan een willekeurige. Als hij uiteindelijk B weer bereikt, is dat hopelijk een snellere route dan eerst. Is het inderdaad een verbetering, dan neemt hij deze nieuwe snelste route weer als basis voor een nieuwe poging. Op deze manier kun je net zo lang doorgaan totdat je er weer een paar honderd verschillende routes zijn geprobeerd. In de meeste gevallen zal dit een snellere route opleveren dan elke keer vanaf A te beginnen.

Parameter Tuning

Het bovenstaande voorbeeld van een navigatiesysteem is een mooi voorbeeld van de kracht van evolutionaire algoritmes, maar dit soort algoritmes hebben ook een bepaalde zwakte. Waarom hebben we bijvoorbeeld voor 50%, 40% en 10% gekozen? Zou 40%, 30%, 30% niet beter werken? Of 91.3%, 4.1%, 4.6%? Een goed antwoord op die vraag is er meestal niet. Over het algemeen worden zulke getallen gewoon gekozen omdat het mooie ronde getallen zijn, of omdat er 10 verschillende percentages zijn geprobeerd en dit nu eenmaal de beste was. Helaas hangt het succes van het algoritme wel heel erg af van deze *parameters*. Kies je te vaak een willekeurige weg, dan worden het nogal vreemde routes. Ga je te vaak richting de snelweg, dan rij je kilometers om als je enkel naar het winkelcentrum wil. En ga je te vaak direct richting je doel, dan kom je terecht op de langzame binnenweggetjes, in plaats van de snelweg. Om te weten wat de beste percentages zijn, moet je alle mogelijke percentages proberen en kijken wat de beste is. Het probleem is daarbij dat er oneindig veel verschillende mogelijkheden zijn, dus het is onmogelijk om ze allemaal af te gaan.

Een oplettende lezer zou nu hebben opgemerkt dat dit exact dezelfde zin

is als hiervoor stond. Daar ging het om het aantal routes tussen A en B, hier om de percentages die gebruikt worden in het algoritme. Het spreekt voor zich dat we hier dus eenzelfde benadering kunnen gebruiken. Zo'n slim algoritme is dan in staat de parameters van een ander algoritme in te stellen, zonder dat algoritme zelfs te kennen. Ze gaan simpelweg op een slimme manier de beste mogelijke parameterwaarden af. Een dergelijk algoritme wordt in deze thesis een *automatische parameter tuning algoritme* of *tuner* genoemd. De fabrikant van het navigatiesysteem zou bijvoorbeeld zo'n tuner kunnen gebruiken om, voordat het systeem in de markt wordt gezet, de beste percentages te bepalen van het route-plan-algoritme. In deze thesis beschrijven we tientallen van dit soort automatische parameter tuning algoritmes, en vergelijken we ze aan de hand van hun verschillende kwaliteiten. Ook introduceren we twee nieuwe methoden: REVAC++ en Bonesa. Deze laatste heeft een aantal unieke kwaliteiten die geen van de andere tuners heeft. We laten dan ook het nut zien van deze kwaliteiten, en tonen aan dat Bonesa behoort tot de absolute top.

Scientific Testing

De bovenstaande manier van parameter tuning wordt in deze thesis *competitive testing* genoemd, omdat het enkel neerkomt op een competitie tussen de verschillende parameter-instellingen. Er wordt niets geleerd over het algoritme, aan het einde weten we nog steeds niet *waarom* iets een goede instelling is, enkel dat het een goede instelling is. Hoewel dit voor sommige toepassingen ook helemaal niet nodig is (een paar voorbeelden daarvan staan ook in deze thesis), is dat verre van een wetenschappelijke aanpak. Stel dat bij de publicatie van dit route-plan-algoritme, bijvoorbeeld door een Amerikaanse wetenschapper, wordt vermeld dat de beste percentages 5%, 94% en 1% zijn, hebben wij daar in Nederland dan wat aan? We willen nog steeds simpelweg van A naar B. Het doel verandert niet, dus er lijkt niets op tegen om deze percentages ook in Nederland te gebruiken. Maar, bij nadere inspectie valt direct de grote waarde op voor richting de snelweg gaan: 94%. Waarom zou dat zo hoog zijn? Wat is daarvan de reden?

Dit is precies het probleem van competitive testing, er is niets geleerd over de parameters. De tuner is blind aan het werk gezet, en de percentages kwamen er vanzelf uit. Hoewel dat al een hele verbetering is ten opzcihte van gewoon 'mooie' getallen gebruiken, is het nog steeds zeer beperkt. Als de maker van Nederlandse navigatiesystemen zomaar deze percentages had overgenomen, had hij namelijk een heleboel boze klanten gehad. In Amerika, waar men grote

afstanden aflegt en snelwegen de verbinding vormen tussen dorpen en steden zijn deze percentages misschien de beste, maar in Nederland met zijn ingewikkelde infrastructuur niet. Als de maker van het route-plan-algoritme de uitkomsten van de parameter tuning had geprobeerd te begrijpen, in plaats van enkel te publiceren, dan was zijn onderzoek ook buiten Amerika van nut geweest. Nu zal geen enkele Nederlandse bouwer zijn algoritme gaan gebruiken, ongeacht hoe goed het ook had kunnen werken als de juiste percentages zouden zijn gebruikt.

Hoewel dit een mooi voorbeeld is van het nut van hetgeen wij *scientific testing* noemen, namelijk het begrijpen en leren over algoritmen, zijn er nog vele andere voorbeelden mogelijk waarin andere aspecten een rol spelen. In het bovenstaande vraagstuk verandert enkel het probleem (van America naar Nederland), maar er zijn nog vele andere manieren waardoor de uitkomsten beinvloed kunnen worden. Is het bijvoorbeeld genoeg als een routeplan algoritme gemiddeld altijd een hele snelle weg vindt? Of moet hij altijd een gemiddeld goede weg vinden? Bij andere vraagstukken zullen aspecten meewegen zoals persoonlijke voorkeuren en het nemen van risico's. Het moge duidelijk zijn dat gewoonweg het algoritme publiceren met de bijbehorende parameters maar van geringe waarde is. Nu computers steeds snellers worden en tuners steeds beter, is er geen enkel excuus meer voor het gebruik van 'mooie' percentages of competitive testing.

In 1995 publiceerde John Hooker voor het eerst over scientific testing, maar zijn werk verdween in het archief door een gebrek aan computerkracht en werd vergeten. Nu 27 jaar later kan dat argument niet meer gebruikt worden en vormt zijn artikel van toen de inspiratie voor deze thesis. Met dit werk proberen we wetenschappers bewust te maken van het nut van competitive, maar vooral scientific testing. We laten verschillende voorbeelden zien waarmee we aantonen dat de huidige methode van gewoonweg mooie getallen gebruiken, onnodig, inefficient en van weinig waarde is. We geven aan op welke aspecten gelet moet worden bij zo'n wetenschappelijke aanpak en hoe de verschillende tuners, zoals het door ons geintroduceerde Bonesa, daarin van waarde kunnen zijn. Het zal een drastische verandering zijn, maar hij is nodig. Daarom eindigt deze thesis met dezelfde hoop en verwachting voor de toekomst die John Hooker 27 jaar geleden al uitsprak:

> "*in plaats van zichzelf te pijnigen over wat de best parameter waarden zijn, doet men wetenschappelijke experimenten om precies vast te kunnen stellen wat hun effect is op de uitkomst.*"
>
> – J.N. Hooker